CS 300 Final Exam Review Fall 2011

This is not a complete list of questions and topics, but a good sampling of questions that will help you study for the final. I strongly advise you to work through every single question.

Review each of your old Exams.
Review each in class Lab.
Review each programming assignment.
Review each set of note and the questions/practice embedded in the notes.

Use the following data structures to answer the questions:

```
typedef struct NODE *NODE_PTR;
typedef struct NODE
{
  char data;
  NODE_PTR psNext;
} NODE;

typedef struct Stack
{
  NODE_PTR psTop;
} Stack;
```

```
typedef struct Tree
{
  struct Tree *psLeft;
  struct Tree *psRight;
  int data;
} Tree;
```

1) The Stack above maintains data using a linked list.
   Write stkPush()
   Write stkPop()

   What is the O ( ) run-time of each of the above functions?

2) Recall the definition of a SET and a BAG. If we were to write one of the two data structures (SET or BAG), and then use that to implement the other data structure, which one should we implement first? (This is similar to building Queue using PriorityQueue).

3) Write the struct definition for the SET and BAG as you defined in question 2.

4) Write insert() for the BAG.

5) Write insert() for the SET.

6) Write contains() for the BAG.

7) Write contains() for the SET.

8) Write your own question

9) Write your own question

```
typedef struct GNode *GNodePtr;
typedef struct GNode
{
  void *pData;
  unsigned int size;
  GNodePtr psNext;
} GNode;
```

10) Write insert() for the linked list created using GNode.

11) Store the following values into a linked list using the insert() function from 10.
"A", 123, 5.1, 'c', "EXAMREVIEW"

12) Write contains() for the linked list created using GNode.

13) Write a function visit() that will take a function pointer and call that function pointer on every node in the linked list.

14) Use the visit() function from 13 to print every node in the linked list that contains a char (You can assume any node with 1 byte of data contains a char).

LINUX

15) What are the Linux command necessary to:
1) change to a directory named cs300Review
2) run an executable file named reviewTest in the current directory
3) capture the output of the executable file hashDriver to a file named hashDriver.out
4) show any differences between the file hashDriver.out and hashDriver.expected.out
5) copy the file hashDriver.out from the local machine to Zeus

MAKEFILES

16) Define Separate Compilation. Give an example.

17) Draw the COMPLETE dependency graph for your Airport's Makefile (and other Makefiles referenced).

POINTER ARITHMETIC

18) Use pointer arithmetic to print every value of the array to the screen.
int *pArray;
int arraySize;
// allocate and fill array.

19) Use pointer arithmetic to print every value of the array to the screen.
int Array[10][5];

HASH TABLES

20) Use Open Address where f(i) = i as the collision handling technique to insert the follow values into a hash table of length 11. The hash function is (N % 11).

Values: 11, 1, 0, 34, 43, 6, 32, 13, 12, 22

Highlight any primary clusters that arise.

21) Use Open Address where f(i) = i^3 as the collision handling technique to insert the follow values into a hash table of length 11. The hash function is (N % 11).

Values: 11, 1, 0, 34, 43, 6, 32, 13, 12, 22

Highlight any primary clusters that arise.
Highlight any secondary clusters that arise.

22) Use Chaining as the collision handling technique to insert the follow values into a hash table of length 11. The hash function is (N % 11).

Values: 11, 1, 0, 34, 43, 6, 32, 13, 12, 22

23) What is the average access time for each element in 20?

24) What is the average access time for each element in 21?

25) What is the average access time for each element in 22?

TREES

26) Insert the following elements into a Binary Search Tree:
100 7 9 1 8 6 120 180 110 121 122

27) Write a function btCountNodes that returns the number of nodes in a Binary Tree.

28) Write a function btLargest that returns the largest value in a Binary Tree.

29) Write a function bstLargest that returns the largest value in a Binary Search Tree.

30) Write a function btPrint that prints the Binary Tree at right in the form:
Parent: X Child: Y Child: Z
Parent: Y Child: A Child: B
Parent: Z Child: Q Child: NULL